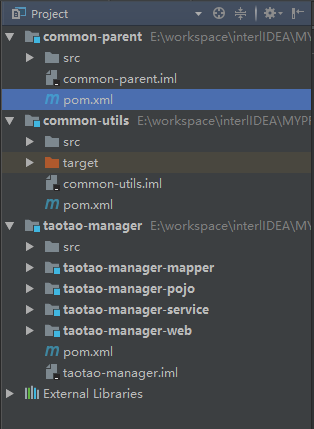
[**基于maven使用IDEA创建多模块项目**](http://blog.csdn.net/williamhappy/article/details/54376855)

鉴于最近学习一个分布式项目的开发，讲一下关于使用IntelliJ IDEA基于Maven创建多模块项目的实际开发，可能有不合适的地方，但是项目是可以跑通的，也请有不足之处，都提出来，一起讨论下。

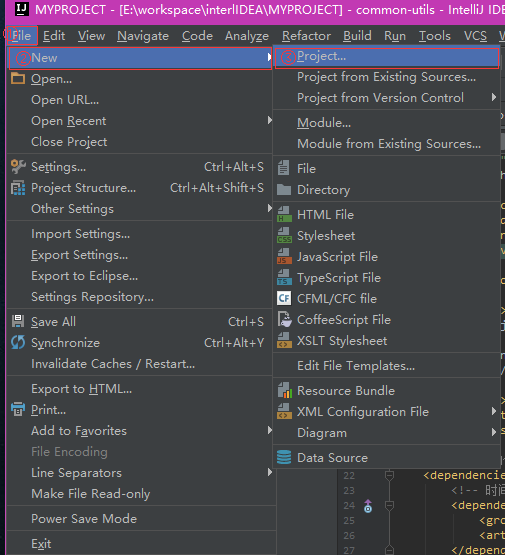
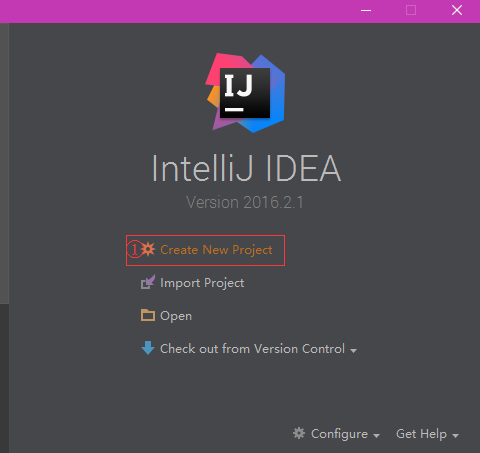
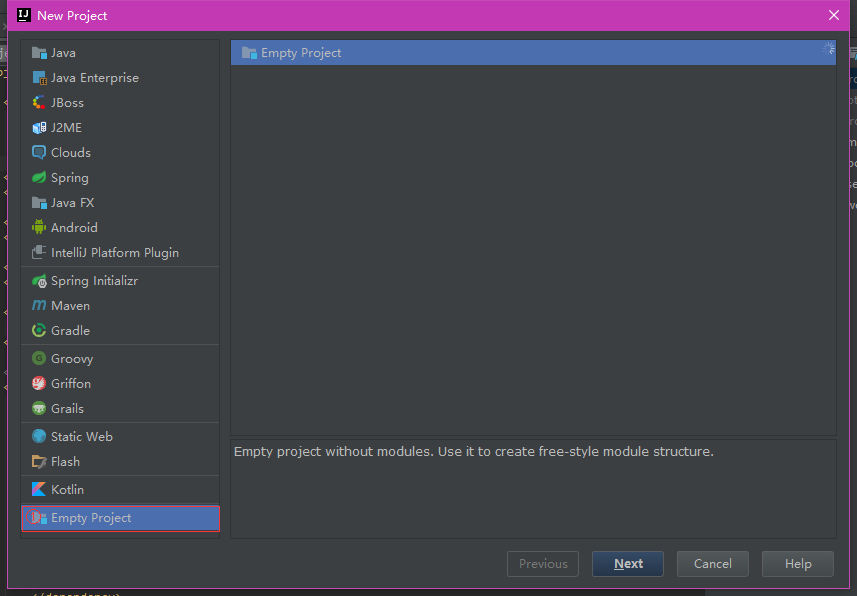
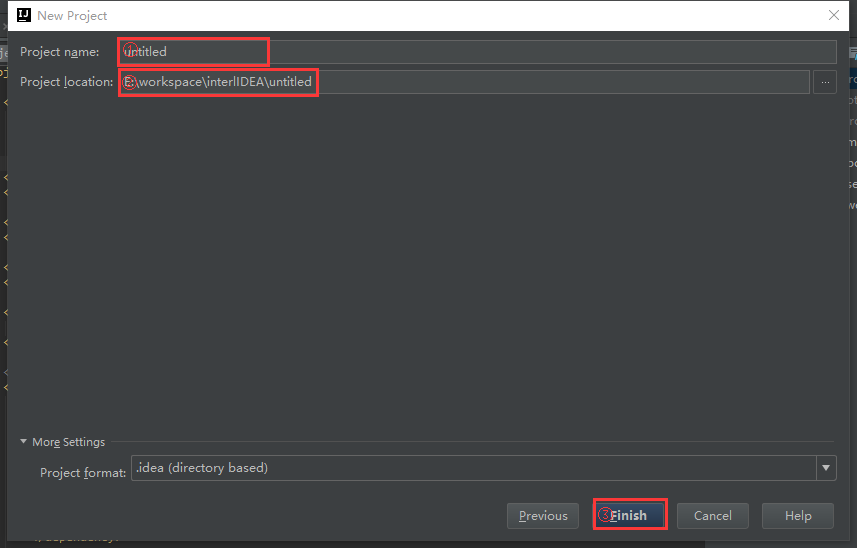
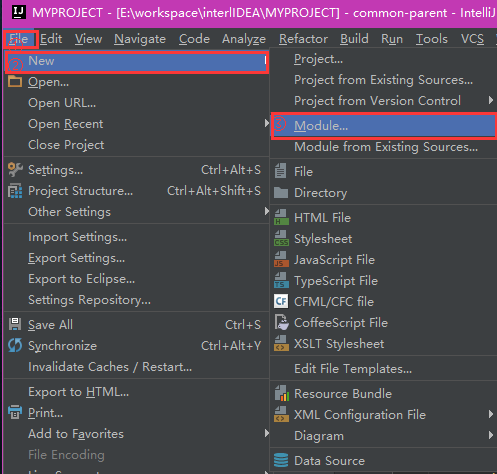
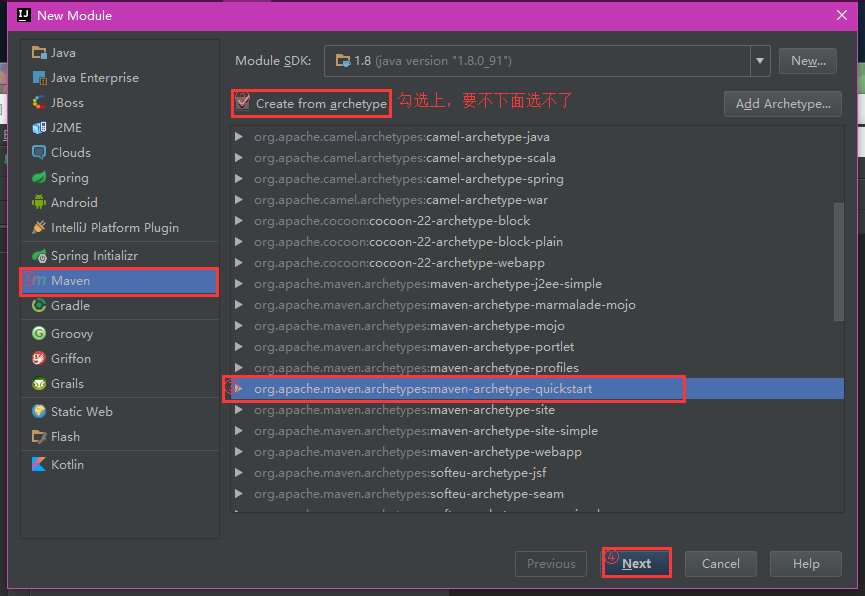
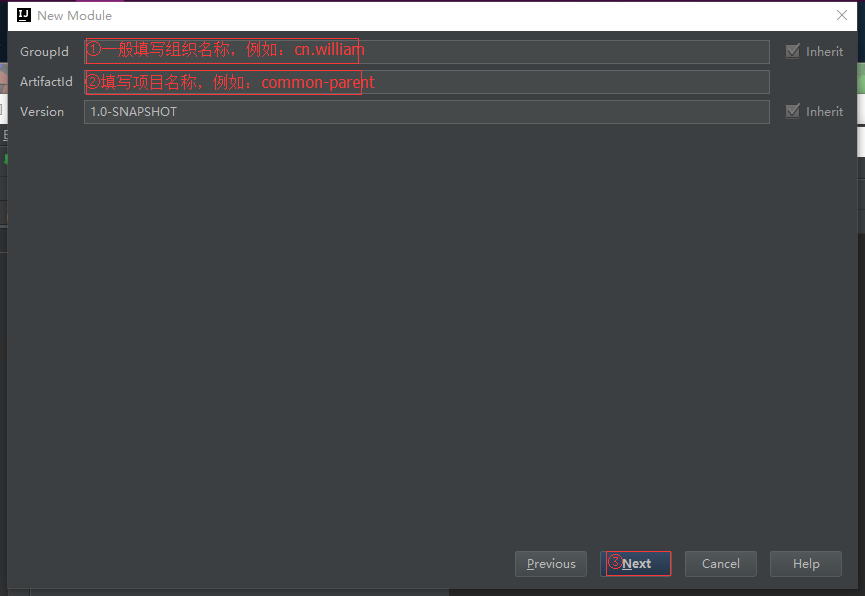
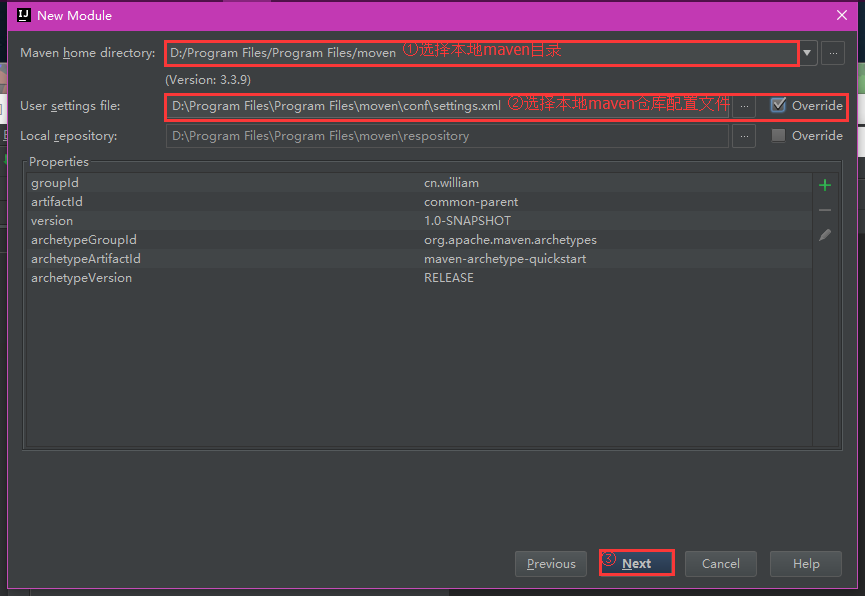
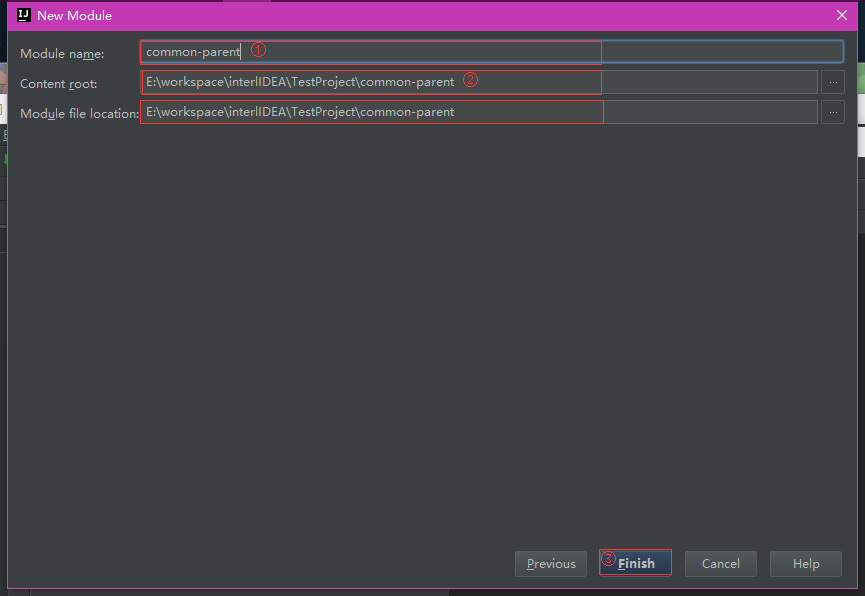
一. 项目工程目录

首先展示一下，最终整个项目的工程目录：



简单介绍一下目录结构：   
common-parent为所有项目的父项目，主要用来管理所有项目使用的jar包及其版本。   
common-utils为公共的工具类项目，继承父项目，它会被打成jar包供其它项目使用。   
taotao-manager为我们自己的项目，继承与我们的父项目。   
taotao-manager-pojo为我们自己项目的一个子模块，依赖与taotao-manager,打成jar包   
taotao-manager-mapper为我们自己项目的一个子模块，依赖与taotao-manager-pojo,打成jar包   
taotao-manager-service为我们自己项目的一个子模块，依赖与taotao-manager-mapper,打成jar包   
taotao-manager-web为我们自己项目的一个子模块，依赖与taotao-manager-service,打成war包

二. 创建父工程

下面为具体操作：   
1. 首先打开IntelliJ IDEA，按下面步骤来：   
   
或者第一次打开IDEA，   
   
2. 来到创建目录，选择Empty project，next   
   
3. 来到New Project页面，填写project name，选择Project location，点Finish   
   
等待IDEA加载完成，进行下一步   
4. 选择File–>New–>Module,进入New Module页面：   
   
5. 在New Module页面，按如下步骤操作：   
   
6. 填写组织名称和项目名称，以及版本号   
   
7. 配置本地maven目录及maven仓库配置文件   
   
8. 创建父工程最后一步，给Module取名，及选择工作目录，选择完成   


等待生成pom文件，注意修改打包方式为pom，修改pom文件如下：(由于pom文件较长，只给出部分，获取所有，请关注[github](http://blog.csdn.net/williamhappy/article/details/54376855))

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>cn.william</groupId>

<artifactId>common-parent</artifactId>

<version>1.0-SNAPSHOT</version>

<packaging>pom</packaging><!--修改打包方式-->

<name>common-parent</name>

<url>http://maven.apache.org</url>

<!--集中定义依赖版本号-->

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<junit.version>4.12</junit.version>

...(省略)

</properties>

<dependencyManagement>

<dependencies>

<!-- 时间操作组件 -->

<dependency>

<groupId>joda-time</groupId>

<artifactId>joda-time</artifactId>

<version>${joda-time.version}</version>

</dependency>

...(省略)

</dependencyManagement>

<build>

<finalName>${project.artifactId}</finalName>

<plugins>

<!-- 资源文件拷贝插件 -->

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-resources-plugin</artifactId>

<version>2.7</version>

<configuration>

<encoding>UTF-8</encoding>

</configuration>

</plugin>

<!-- java编译插件 -->

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.5.1</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

<encoding>UTF-8</encoding>

</configuration>

</plugin>

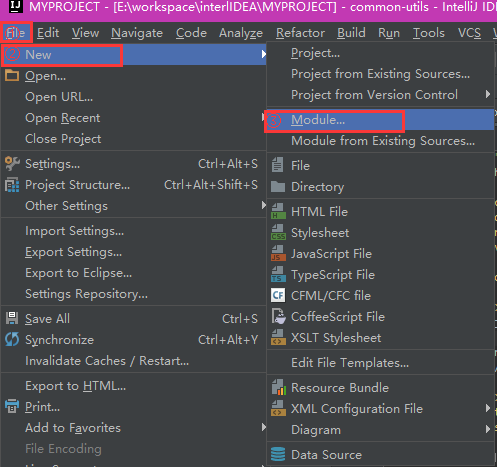
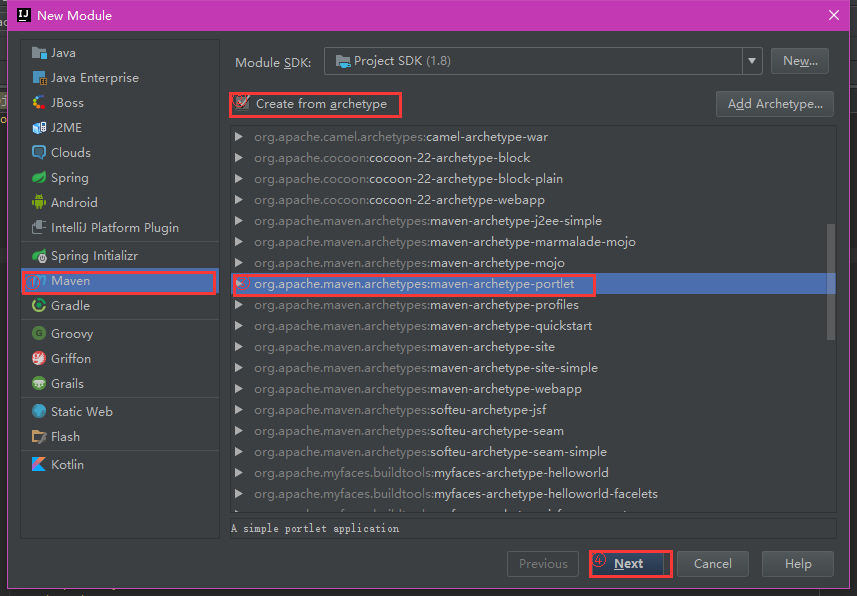
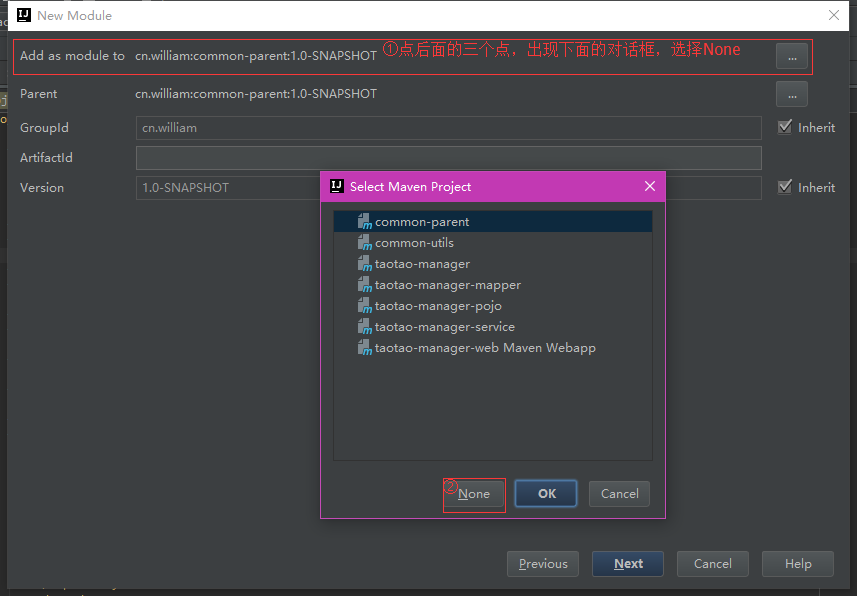
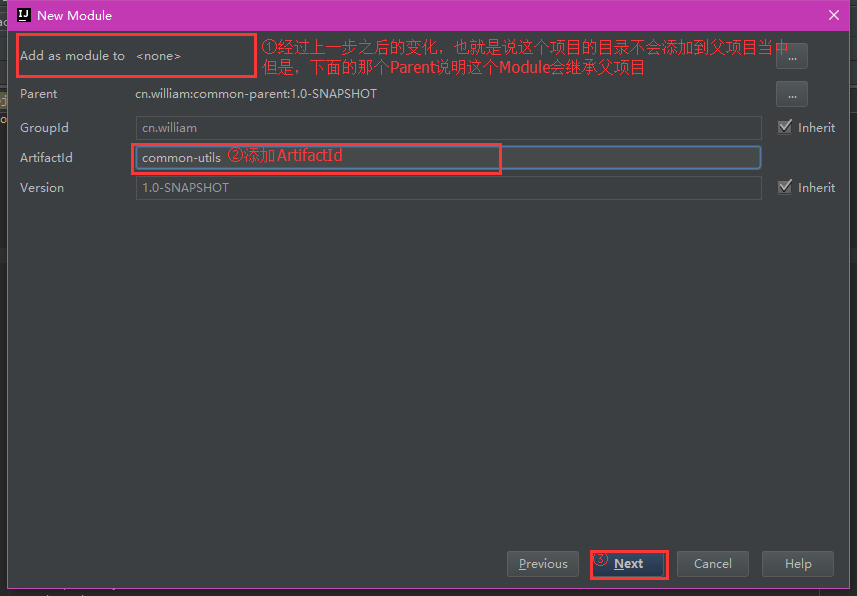
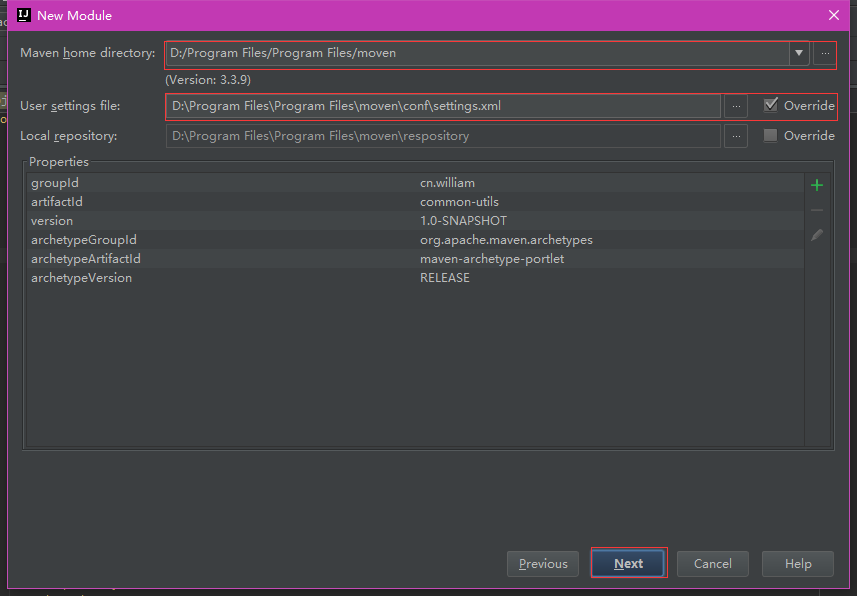
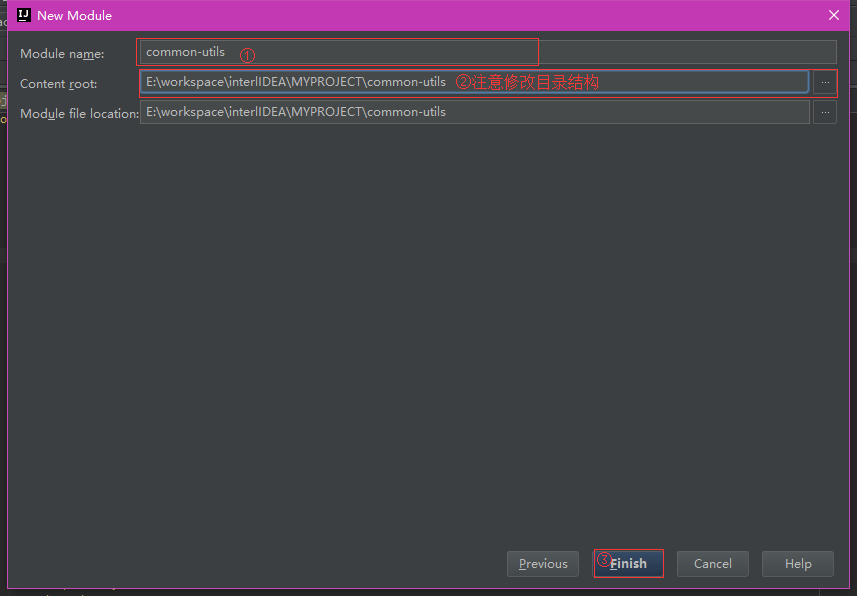
</plugins>

</build>

</project>

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37
* 38
* 39
* 40
* 41
* 42
* 43
* 44
* 45
* 46
* 47
* 48
* 49
* 50
* 51
* 52
* 53
* 54
* 55
* 56
* 57
* 58

三. 创建公共工具类

首先，公共工具类是继承自父项目common-parent的。   
需要注意的一点是一点，为了项目看起来为分模块的形式（类似在eclipse中开发），在创建common-utils中有一步需要注意：   
   
   
下一步特别注意，这一步会影响目录结构，   
   
   
   


同样注意项目的打包方式为jar，和父项目有所不同。

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<parent>

<artifactId>common-parent</artifactId>

<groupId>cn.william</groupId>

<version>1.0-SNAPSHOT</version>

<relativePath>../common-parent/pom.xml</relativePath>

</parent>

<modelVersion>4.0.0</modelVersion>

<artifactId>common-utils</artifactId>

<packaging>jar</packaging><!-- 打成jar包 -->

<name>common-utils</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<!-- jar包的依赖 -->

<dependencies>

<!-- 时间操作组件 -->

<dependency>

<groupId>joda-time</groupId>

<artifactId>joda-time</artifactId>

</dependency>

...(省略)

</dependencies>

</project>

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32

三. 创建开发项目

创建开发项目taotao-manager，其创建步骤同上，同样是继承common-parent,同样修改其pom文件，详细如下：

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<parent>

<artifactId>common-parent</artifactId>

<groupId>cn.william</groupId>

<version>1.0-SNAPSHOT</version>

<relativePath>../common-parent/pom.xml</relativePath>

</parent>

<modelVersion>4.0.0</modelVersion>

<artifactId>taotao-manager</artifactId>

<packaging>pom</packaging>

<name>taotao-manager</name>

<url>http://maven.apache.org</url>

<modules>

<module>taotao-manager-pojo</module>

<module>taotao-manager-mapper</module>

<module>taotao-manager-service</module>

<module>taotao-manager-web</module>

</modules>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<!-- 依赖管理 -->

<dependencies>

<dependency>

<groupId>cn.william</groupId>

<artifactId>common-utils</artifactId>

<version>1.0-SNAPSHOT</version>

</dependency>

</dependencies>

<build>

<!-- 配置插件 -->

<plugins>

<plugin>

<groupId>org.apache.tomcat.maven</groupId>

<artifactId>tomcat7-maven-plugin</artifactId>

<configuration>

<port>8080</port>

<path>/</path>

</configuration>

</plugin>

</plugins>

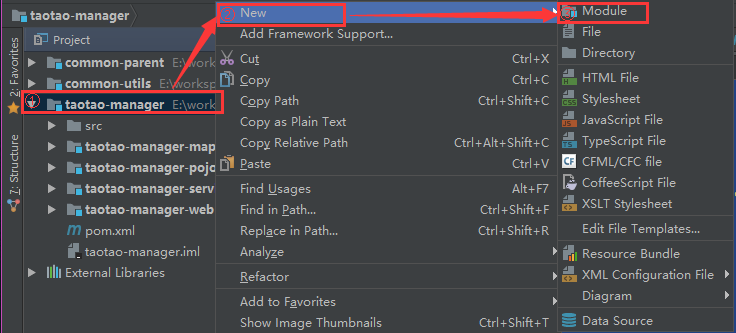
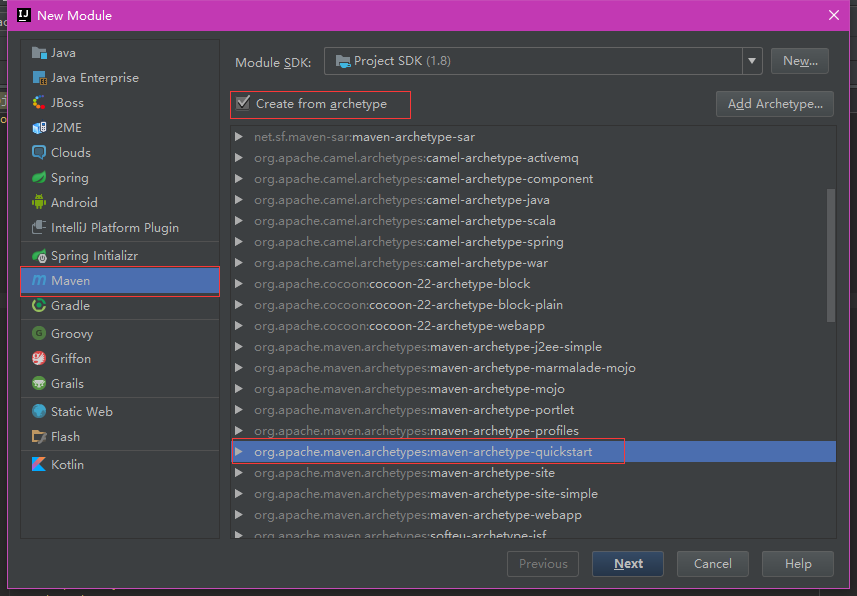
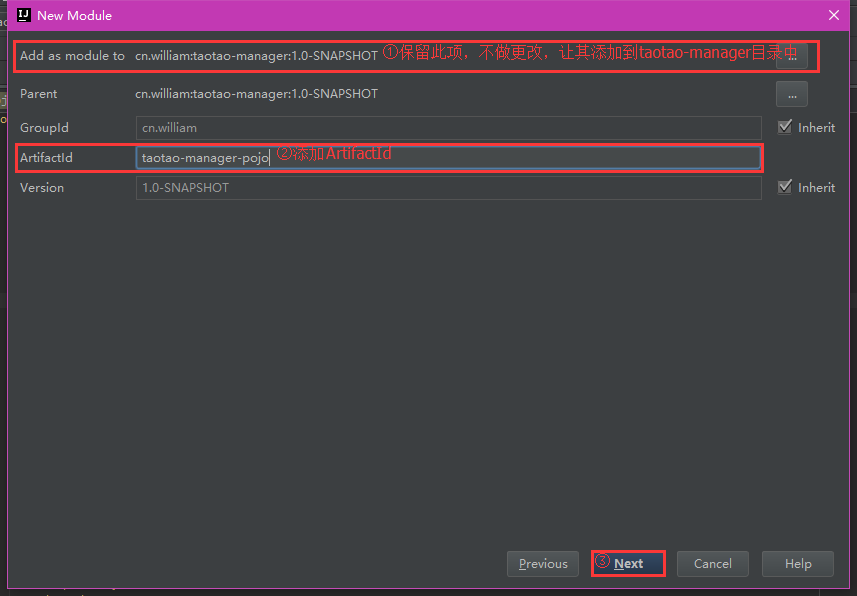
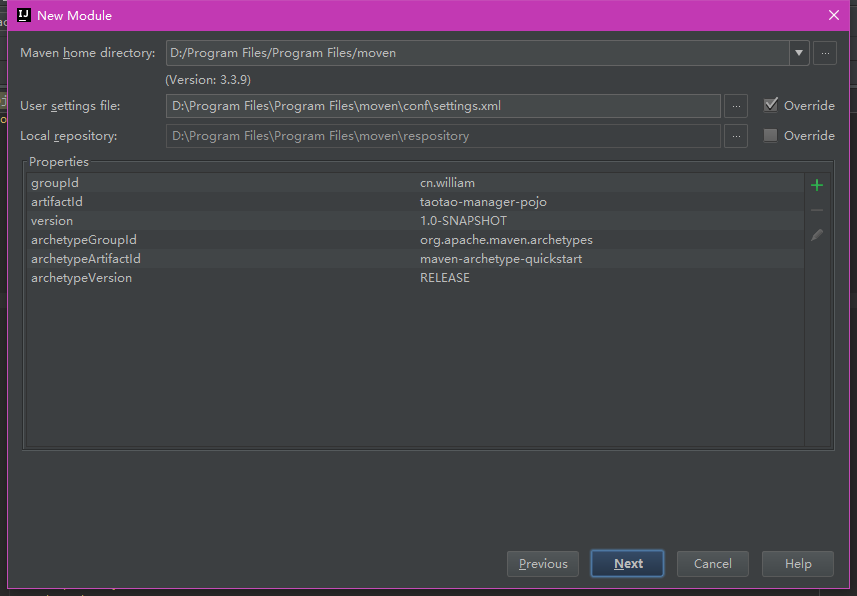
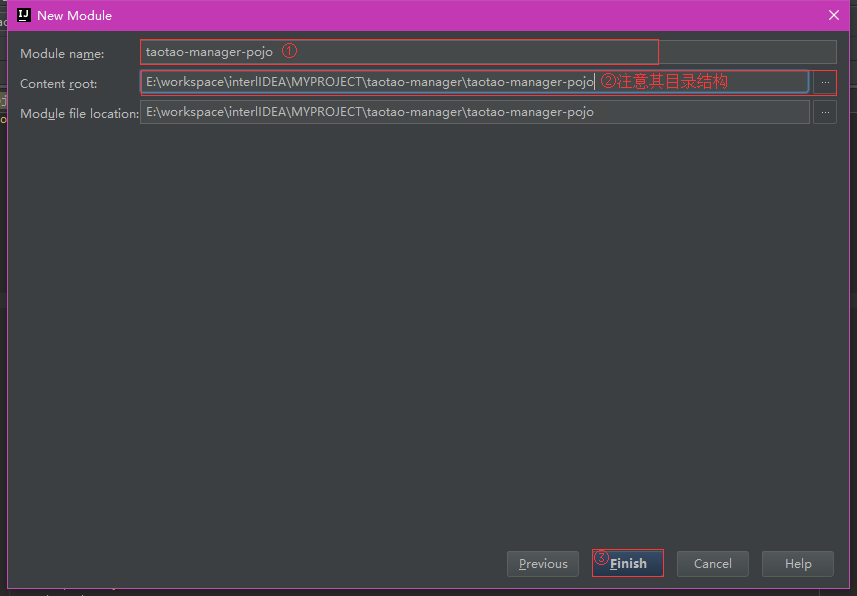
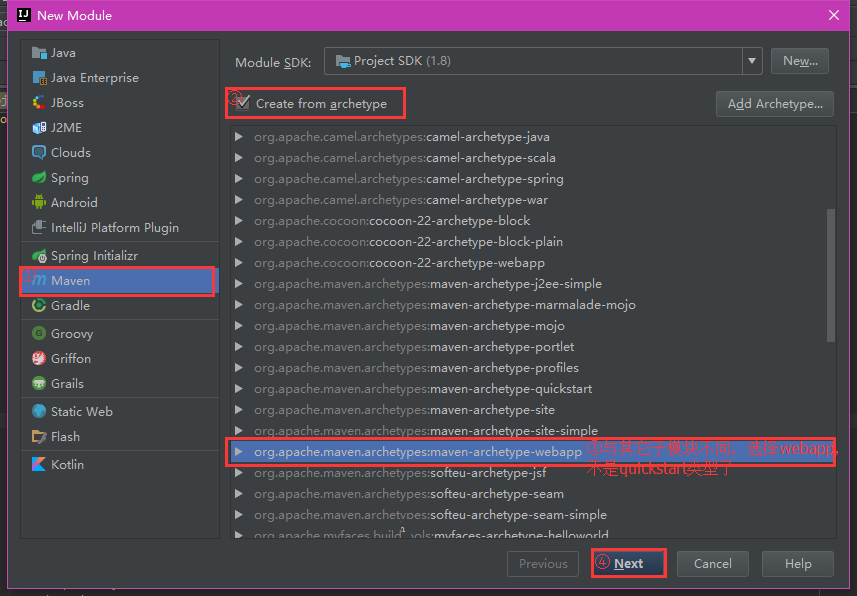
</build>

</project>

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37
* 38
* 39
* 40
* 41
* 42
* 43
* 44
* 45
* 46
* 47
* 48
* 49
* 50
* 51
* 52

四. 创建开发项目子模块

创建taotao-manager-pojo，mapper，service，web等子模块，详细如下：

1. 右键点击taotao-manager，选择New，Module，   
      
      
   这一步注意，与创建common-utils时不同，是将其子模块添加到taotao-manager目录下的：   
      
   maven的相关配置，保持默认：   
      
   注意子模块所在目录就好：   
   
2. 创建taotao-manager下其它子模块   
   上面说的时创建taotao-manager-pojo子模块，另外三个子模块的创建方式与其类似，有一点不同时，在创建taotao-manager-web子模块时，maven项目的类型选择如下：   
   
3. 修改子模块的pom文件

子模块pojo，mapper，service的打包方式全为jar，只有web子模块打包为war包。

pojo子模块，pom.xml详细如下：

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<parent>

<artifactId>taotao-manager</artifactId>

<groupId>cn.william</groupId>

<version>1.0-SNAPSHOT</version>

</parent>

<modelVersion>4.0.0</modelVersion>

<artifactId>taotao-manager-pojo</artifactId>

<packaging>jar</packaging>

<name>taotao-manager-pojo</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

</project>

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21

mapper子模块依赖与pojo子模块，pom.xml详细如下：

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<parent>

<artifactId>taotao-manager</artifactId>

<groupId>cn.william</groupId>

<version>1.0-SNAPSHOT</version>

</parent>

<modelVersion>4.0.0</modelVersion>

<artifactId>taotao-manager-mapper</artifactId>

<packaging>jar</packaging>

<name>taotao-manager-mapper</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<dependencies>

<!-- 依赖与pojo子模块 -->

<dependency>

<groupId>cn.william</groupId>

<artifactId>taotao-manager-pojo</artifactId>

<version>1.0-SNAPSHOT</version>

</dependency>

<!-- Mybatis -->

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis</artifactId>

</dependency>

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis-spring</artifactId>

</dependency>

<dependency>

<groupId>com.github.miemiedev</groupId>

<artifactId>mybatis-paginator</artifactId>

</dependency>

<dependency>

<groupId>com.github.pagehelper</groupId>

<artifactId>pagehelper</artifactId>

</dependency>

<!-- MySql -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

</dependency>

<!-- 连接池 -->

<dependency>

<groupId>com.alibaba</groupId>

<artifactId>druid</artifactId>

</dependency>

</dependencies>

</project>

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37
* 38
* 39
* 40
* 41
* 42
* 43
* 44
* 45
* 46
* 47
* 48
* 49
* 50
* 51
* 52
* 53
* 54
* 55
* 56
* 57

service子模块依赖与mapper子模块，pom.xml详细如下：

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<parent>

<artifactId>taotao-manager</artifactId>

<groupId>cn.william</groupId>

<version>1.0-SNAPSHOT</version>

</parent>

<modelVersion>4.0.0</modelVersion>

<artifactId>taotao-manager-service</artifactId>

<packaging>jar</packaging>

<name>taotao-manager-service</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<dependencies>

<!-- 依赖与mapper子模块 -->

<dependency>

<groupId>cn.william</groupId>

<artifactId>taotao-manager-mapper</artifactId>

<version>1.0-SNAPSHOT</version>

</dependency>

<!-- Spring -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-beans</artifactId>

</dependency>

...（省略）

</dependencies>

</project>

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37
* 38
* 39

web子模块依赖与service子模块，pom.xml详细如下：

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd">

<parent>

<artifactId>taotao-manager</artifactId>

<groupId>cn.william</groupId>

<version>1.0-SNAPSHOT</version>

</parent>

<modelVersion>4.0.0</modelVersion>

<artifactId>taotao-manager-web</artifactId>

<packaging>war</packaging><!--打成war包-->

<name>taotao-manager-web Maven Webapp</name>

<url>http://maven.apache.org</url>

<dependencies>

<!-- 依赖于service -->

<dependency>

<groupId>cn.william</groupId>

<artifactId>taotao-manager-service</artifactId>

<version>1.0-SNAPSHOT</version>

</dependency>

...（省略）

</dependencies>

<build>

<finalName>taotao-manager-web</finalName>

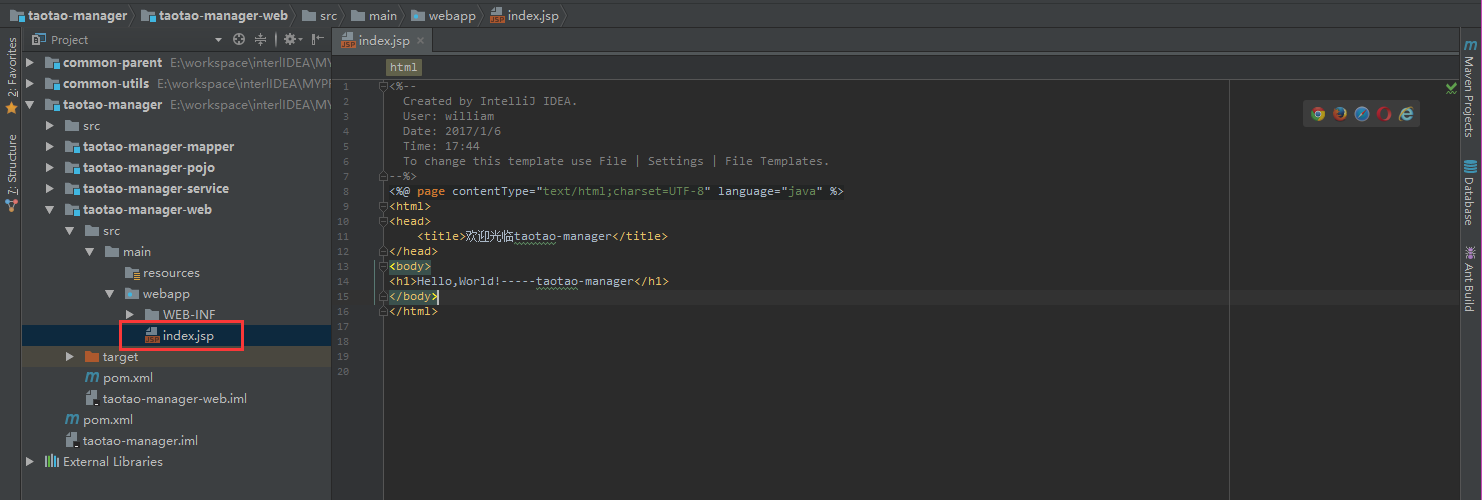
</build>

</project>

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26

关于上面省略的部分，可查看github完整项目，地址：

五. 运行项目

1. 创建index.jsp   
   至此，项目创建已经完成，我们可以在taotao-manager-web的webapp目录下创建index.jsp,详情如下：   
   
2. 使用tomcat插件   
   我们使用maven的tomcat插件来运行项目，在此之前，确保在taotao-manager的pom文件中配置了tomcat的插件：

...（省略）

<build>

<!-- 配置插件 -->

<plugins>

<plugin>

<groupId>org.apache.tomcat.maven</groupId>

<artifactId>tomcat7-maven-plugin</artifactId>

<configuration>

<port>8080</port>

<path>/</path>

</configuration>

</plugin>

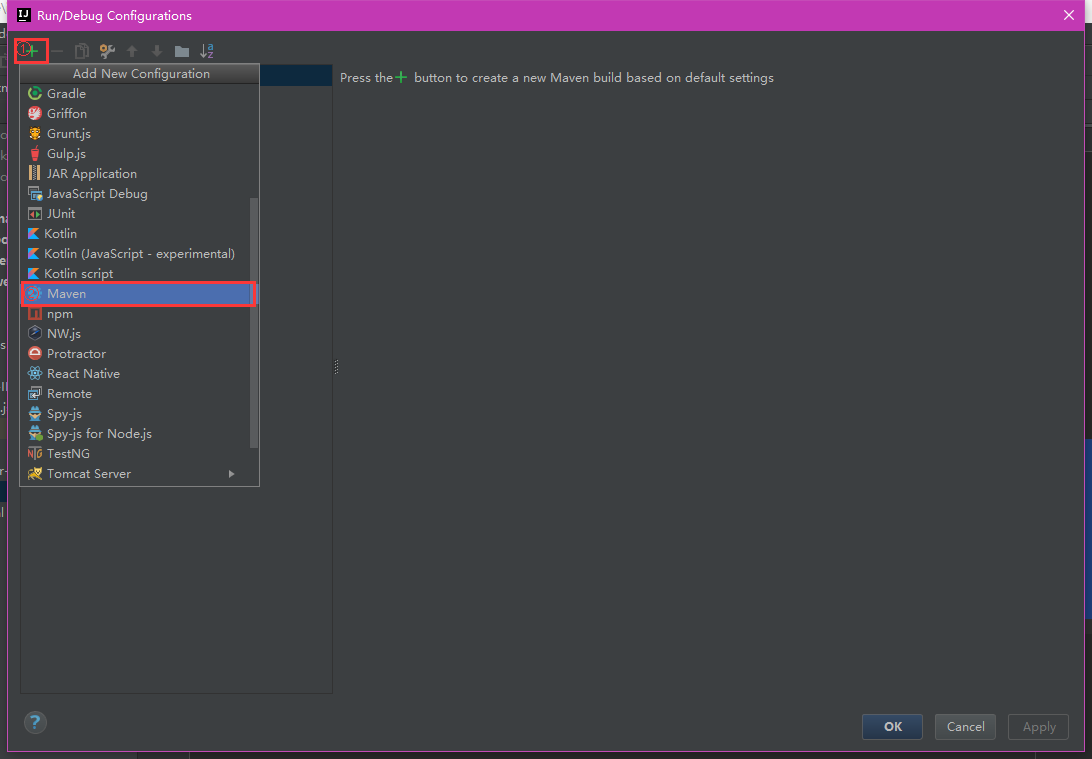
</plugins>

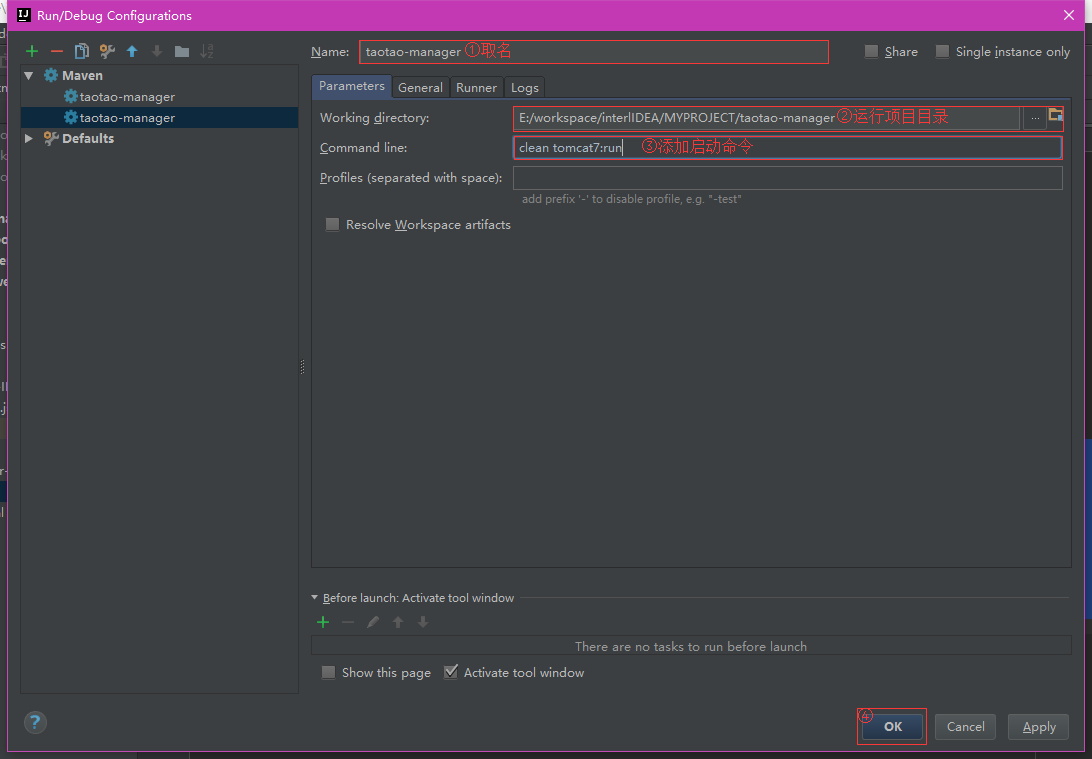
</build>

...（省略）

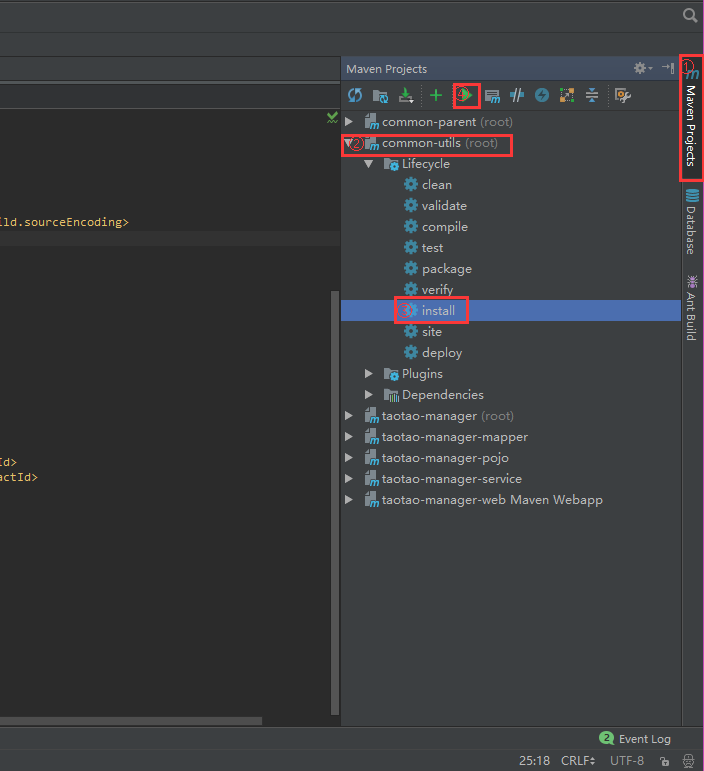
* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15

1. 配置Maven Tomcat Plugin运行web项目

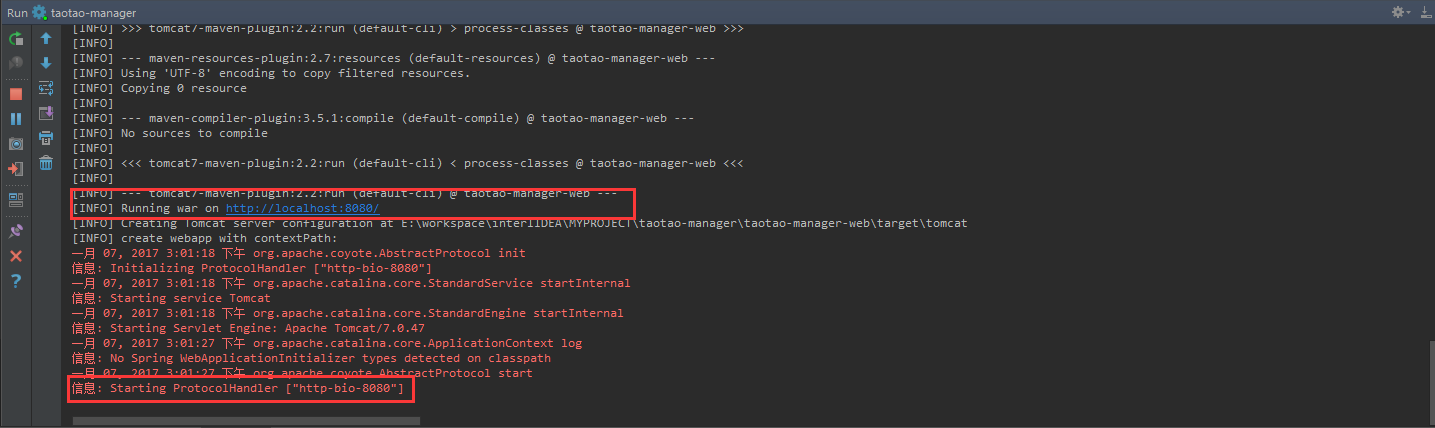
（1） 添加新配置   


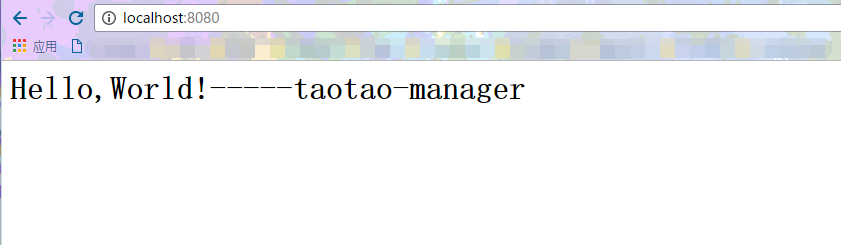


（2） 点击运行，或者点击旁边的debug   
tomcatRun

（3） 出项错误   
出现错误，可能是common-parent和common-utils没有安装到本地仓库，出现的错误是找不到common-utils的jar包，那就需要我们安装一下。   


等待将jar包安装到本地仓库，稍微有点慢可能，耐心等。。。

安装完成之后，再次运行，出现下面的信息，证明运行成功：   


通过浏览器访问：   


至此，我们的项目环境搭建成功，接下来就是实际的项目开发了。

声明：   
① 关于上面代码中出现的省略的部分，是由于内容偏长，所以省略，想要查看完成项目，请查看本人的github，地址：   
② 关于上面有任何不足之处，敬请各位谅解，毕竟本人技术有限，也在学习阶段，也可以给出合理意见，共同改进。   
③ 黑黑，喜欢的，加个关注白，哈哈哈